

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of:

Hahn

Serial No.: 10/726,948

Filed: December 2, 2003

For: **USER-SPACE RESOURCE
MANAGEMENT**

Confirmation No. 5353

Art Unit: 2195

Examiner: Wai, Eric Charles

Atty Docket No. SUNP8475-
US-NP

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF UNDER 37 CFR § 41.37

I. REAL PARTY IN INTEREST

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
USA

II. RELATED APPEALS AND INTERFERENCES

No other appeals or interferences are currently known to Appellant that will directly affect, be directly affected by, or have a bearing on the decision to be rendered by the Board of Patent Appeals and Interferences in the present appeal.

III. STATUS OF CLAIMS

Claims 9, 12, 14-23, 25 and 36-40 are pending in the application. Claims 1-8, 10, 11, 13, 24, 26-35 and 41-53 have been canceled. The rejection of claims 9, 12, 14-23, 25 and 36-40 is the subject of this appeal.

IV. STATUS OF AMENDMENTS

No claim amendments were filed after the mailing of the Final Office Action on November 12, 2009 (“Final Office Action”). All claim amendments have been entered, and pending claims 9, 12, 14-23, 25 and 36-40 are provided in the attached Claims Appendix.

V. SUMMARY OF CLAIMED SUBJECT MATTER

Independent Claim 9:

Independent claim 9 is directed to a computer implemented method including “monitoring usage of a computing resource utilized by a workload wherein said monitoring is performed by a process within a user space and the process monitors the user space only.” A representative computer system 100 usable in conjunction with the method of independent claim 9 is shown in Figure 1 that illustrates an operating system space 180 and a user space 110. The user space 110 may include one or more workloads 111-115, each of which may include one or more processes 121-123 and may be allocated computing resources (e.g., physical memory). See page 13, line 17 through page 14, line 16 of Appellant’s specification. Computer resource usage by the workload is monitored because it is typically a limited resource. See page 3 at lines 9-12.

Moreover, the monitoring of independent claim 9 “is performed within a user space and the process monitors the user space only.” See page 13 at lines 3-4 and 11-12 and page 15 at lines 9-10. Performing the monitoring within the user space advantageously provides significant benefits in operating system scalability and efficiency. See page 18, lines 10-21. For instance, implementing computer resource (e.g., physical memory) monitoring outside of a kernel (e.g., within a user space) enables the physical memory usage monitoring to operate less frequently thus consuming fewer computing resources. See page 18, line 22 through page 19, line 10.

In independent claim 9, the workload “includes a plurality of running processes, the plurality of running processes are a subset of all processes that are running in the user space.” As seen in Figure 1, the workload 112 includes a plurality of running processes 121-123 which are a subset of all processes running (e.g., processes 121, 122 running in other workloads 111, 113-115) in the user space 110.

The method of independent claim 9 also includes “responsive to exceeding a limit on utilization of said computing resource, decreasing usage of said computing resource by said workload, said computing resource comprises physical memory.” See flowchart in Figure 3 which is described starting on page 3 of Appellant’s specification. Moreover, in the method of independent claim 9, “said decreasing usage of said computing resource comprises paging a portion of said physical memory assigned to said workload out of said physical memory and said decreasing usage does not halt operation of said workload.” “Paging” is discussed at page 4, lines 7-13. Generally, paging serves to decrease the amount of physical memory utilized by a workload. See page 21 at lines 11-13. Avoiding any stoppage or interruption of the workload advantageously allows the workload to continue even as workload resources are decreased. See page 21, line 16 through page 22, line 2 and page 24, line 19 through page 25, line 3 of Appellant’s specification.

Independent Claim 16:

Independent claim 16 is directed to “a computer implemented method for memory management of a workload from within the workload.” See representative computer system in Figure 1 and flow diagram of Figure 4. The method includes “accessing a list of memory pages assigned to said workload in a user space, the workload includes a plurality of running processes, the plurality of running processes are a subset of all processes that are running in the user space.” See box 410 in Figure 4. The list may be created and/or maintained by the operating system and may be accessed by a process (e.g., resource usage monitor) executing within the workload. See page 23, lines 1-6. As seen in Figure 1, the workload 112 includes a plurality of running processes 121-123 which are a subset of all processes running (e.g., processes 121, 122 running in other workloads 111, 113-115) in the user space 110.

Thereafter, the method of independent claim 16 includes “responsive to a request from a first process of said workload for memory which exceeds a predetermined memory limit for said workload, selecting a plurality of memory pages from said list of memory pages, wherein the plurality of memory pages includes least recently used memory pages assigned to the workload.” See box 420 in Figure 4 and page 23, lines 8-12 of Appellant’s specification. It may be useful to monitor workload computer resource usage (e.g., when memory usage exceeds a memory limit)

as it is typically a limited resource. See page 3 at lines 9-12. When workload usage has exceeded such a limit, it may be advantageous to select least recently used pages because such pages may have the least impact upon workload processes. See page 23 at lines 14-18.

The method also includes “initiating a second process within the user space to page out said plurality of memory pages wherein at least a portion of said workload continues to operate subsequent to said initiating.” See box 430 in Figure 4. For instance, the second process (e.g., an “Agent LWP” process) can be initiated from within the target workload and can make an operating system call to page out the selected or identified pages. See page 24, lines 12-17. Moreover, allowing at least a portion of the workload to operate after the initiating step beneficially allows at least some use of the workload during the paging out process which can enhance system performance. See page 24, line 19 through page 25, line 3.

Independent Claim 36:

Independent claim 36 is directed to a computer implemented method including “accessing memory usage for a workload and examining page usage for each process of said workload, the workload exists within a user space and includes a plurality of running processes, the plurality of running processes are a subset of all processes that are running in the user space” and “aggregating usage of said each process to determine an aggregate usage for said workload.” See box 610 in Figure 6 of Appellant’s specification. For instance, the workload may be a specific workload (e.g., workload 112 in Figure 1) of a set of workloads (e.g., workloads 111 and 113-115 in Figure 1).

Thereafter in independent claim 36, “if said aggregate usage does not exceed a memory utilization limit for said workload” (see diamond 620), the method includes “repeating said accessing and aggregating for a next workload.” See box 625 in Figure 6. However, “if said aggregate usage exceeds said memory utilization limit for said workload,” the method includes “determining least recently used pages based on accessed bits associated with said workload.” See box 630 in Figure 6. As discussed previously, use of least recently used pages rather than use of more frequently used pages reduces system impact during a paging procedure as such

pages are used relatively infrequently anyway. Bits can be associated with a “scoreboard” and be indicative of how frequently used a page is. See page 25, lines 16-17.

Further “if said aggregate usage exceeds said memory utilization limit for said workload,” the method includes “supplying a range of least recently used pages in a system call to an operating system kernel for evicting said range of least recently used pages to reduce resource usage by said workload.” See box 640 in Figure 6 and page 27, lines 1-3. Also, the method includes “retaining at least partial operation of said workload during said page evicting such that the at least partial operation does not stop said workload.” See box 650 in Figure 6 and page 27, lines 5-6. By allowing at least a portion of the workload to operate after the initiating step, at least some use of the workload remains during the paging out process which can enhance system performance. See page 24, line 19 through page 25, line 3.

Independent Claim 38:

Independent claim 38 is directed to “a computer implemented method of managing computer resources over a plurality of workloads.” The method includes “for each workload of said plurality of workloads, monitoring respective workload resource usage against a respective allotment of each workload.” A representative computer system 100 usable in conjunction with the method of independent claim 38 is shown in Figure 1 that illustrates a user space 110 with one or more workloads 111-115, each of which may include one or more processes 121-123 and may be allocated computing resources (e.g., physical memory). See page 13, line 17 through page 14, line 16 of Appellant’s specification. For instance, a workload monitor system 140 may monitor workload resource usage against an allotment of each workload, the allotment being represented by memory utilization gauges 130.

Thereafter, the method of independent claim 38 includes “determining a range of computer resources to page out for each workload whose resource usage exceeds its respective allotment.” For instance, a plurality of “least recently-used pages” may be paged out. See page 15, line 22 through page 16, line 1. Moreover, the method includes “initiating a paging out operation of said range of computer resources and wherein said monitoring, said determining and said initiating all occur within a process of user space.” See box 430 of Figure 4, page 13, lines

2-4, page 15, lines 9-21 and page 16, lines 1-3. For instance, implementing such computer resource (e.g., physical memory) monitoring, determining and initiating outside of a kernel (e.g., within a user space) enables the physical memory usage monitoring to operate less frequently thus consuming fewer computing resources. See page 17, lines 5-8 and page 18, line 22 through page 19, line 10.

Independent claim 38 also includes “wherein each of the plurality of workloads exists within a user space and includes a plurality of running processes, the plurality of running processes are a subset of all processes that are running in the user space, and paging out said range of computer resources.” As seen in Figure 1, the workload 112 includes a plurality of running processes 121-123 which are a subset of all processes running (e.g., processes 121, 122 running in other workloads 111, 113-115) in the user space 110. Moreover, “each workload whose resource usage exceeds its respective allotment remains partially operable during said paging out of its respective range of computer resources.” Allowing at least a portion of the workload to remain operable advantageously allows at least some use of the workload during the paging out operation which can enhance system performance. See page 24, line 19 through page 25, line 3.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1) Whether claims 9, 12, 14-23, 25 and 38-40 are unpatentable under 35 U.S.C. 103(a) over Applicant’s admitted prior art (“AAPA”) in view of “Workload management-many questions”, IT Resource Forums, 7-10-02 to Uri (“Uri”), and further in view of U.S. Patent No. 4,912,629 to Shuler, Jr. (“Shuler”).

2) Whether claims 36 and 37 are unpatentable under 35 U.S.C. 103(a) over “Modern Operating Systems”, 2nd Edition, 2001 to Tanenbaum (“Tanenbaum”) in view of AAPA, and further in view of Shuler.

VII. ARGUMENTS

Rejection of Claims 9, 12, 14-23, 25 and 38-40 under 35 U.S.C. §103(a) Based on AAPA, Uri and Shuler is Improper

In the Final Office Action, claims 9, 12, 14-23, 25 and 38-40 were rejected under 35 U.S.C. §103(a) as being unpatentable over AAPA in view of Uri, and further in view of Shuler. This rejection is traversed because AAPA, Uri and Shuler do not alone or in any combination disclose at least one limitation in independent claims 9, 16 and 38.

In response to the Final Office Action, Appellant appealed the rejection of these claims such that there is no Advisory Action. The following discussion will address the Examiner's claim rejections and "Response to Arguments" section from the Final Office Action.

Independent Claim 9:

Independent claim 9 is directed to a computer implemented method including monitoring usage of a computing resource utilized by a workload, the monitoring being performed by a process within a user space and the process monitoring the user space only. See page 13 at lines 3-4 and 11-12 and page 15 at lines 9-10. The workload includes a plurality of running processes which are a subset of all processes that are running in the user space. As seen in Figure 1, the workload 112 includes a plurality of running processes 121-123 which are a subset of all processes running (e.g., processes 121, 122 running in other workloads 111, 113-115) in the user space 110. The method also includes, responsive to exceeding a limit on utilization of the computing resource, decreasing usage of the computing resource by the workload, wherein the computing resource includes physical memory. See flowchart in Figure 3. Moreover, the decreasing usage step includes paging a portion of the physical memory assigned to the workload out of the physical memory and the decreasing usage does not halt operation of the workload. AAPA, Uri and Shuler do not alone or in any combination disclose or suggest the method of independent claim 9.

1) AAPA, Uri and Shuler do not teach "wherein said monitoring is performed by a process within the user space and the process monitors the user space only" as recited in independent claim 9

AAPA generally discusses how a typical operating system response to physical memory requirements exceeding the physical memory available in a computer system is to copy portions of the physical memory to storage (“paging”). See middle of page 4 of Appellant’s specification. Conventional methods of selecting particular pages for “paging” include those that are “least recently used” and/or pages from those workloads that have exceeded their memory allocation. See bottom of page 4 to middle of page 5 of Appellant’s specification. Selecting pages from a given workload when the workload has exceeded its memory allocation is implemented as an operating system kernel process. See page 5, line 13 of Appellant’s specification.

However and as admitted by the Examiner, AAPA does not at least disclose “wherein said monitoring is performed by a process within the user space and the process monitors the user space only” (see middle of page 3 of Final Office Action). After asserting “Uri teaches using Memory Resource groups wherein a block of memory is allocated to each workload and a separate memory management subsystem will be created for each workload, i.e. part of that workload (pg 3 paragraph 2),” the Examiner contended that it would have been obvious to perform the “accessing” within the user space to allow for more fine grained control and for resources to be allocated based on business priorities. See bottom of page 3 to top of page 4 of Final Office Action. Even though independent claim 9 does not recite an “accessing” step, Appellant will assume for purposes of this Appeal Brief that the Examiner intended “monitoring”. Nevertheless, Appellant respectfully asserts that Uri does not disclose or suggest a process that monitors usage of a computing resource by a workload within a user space and that monitors the user space only as recited in independent claim 9.

In contrast, Uri discusses in-kernel memory controls known as “Memory Resource Groups” that allow a block of memory to be allocated to each PRM (e.g., Process Resource Manager) group and a memory management subsystem to be created for each group. See paragraph 2, lines 10-13 on page 3 of Uri. The Examiner essentially likened the memory management subsystem being created for each group to the monitoring process of independent claim 9 being performed within a user space. See bottom of page 3 of Final Office Action where Examiner states “i.e. part of that workload”. Appellant respectfully disagrees with the Examiner’s contention. Actually, allocating a memory management subsystem for each PRM

group as in Uri does not necessarily require any memory monitoring processes to be performed within a user space as in independent claim 9. More specifically, the Memory Resource Groups and memory management subsystems were created at least in part to address the situation where it may be desirable to not stop certain processes even in spite of using too much memory. See paragraph 2, lines 7-10 of Uri. As a result, the memory management subsystems likely would be implemented *outside* of a user space to avoid halting processes.

Furthermore, the Memory Resource Groups and memory management subsystems are implemented as *in-kernel memory controls*. See paragraph 2, lines 10-11 of Uri. Conversely, independent claim 9 recites that the monitoring is performed by a process *within a user space* (which Appellant's specification clearly defines as being *different* than in-kernel processes). Thus, the system of Uri provides an opposite technique from the method of independent claim 9 at least in this regard. Performing the monitoring *within the user space* as in independent claim 9 advantageously allows, for instance, physical memory usage monitoring to operate less frequently (compared to when such monitoring is performed as a kernel process as in Uri) thus consuming fewer computing resources. See page 18, line 22 through page 19, line 10 of Appellant's specification. The combination of AAPA and Uri fails to show or suggest the monitoring step of independent claim 9 (and Shuler is not cited for any teaching relevant to the recited monitoring step). In contrast, combining AAPA and Uri would result in a method of implementing a process of paging out pages of a workload from memory (from AAPA) that is implemented as an "in-kernel" process (from Uri) which is distinct from the method of independent claim 9. For at least this reason, Appellant respectfully requests that independent claim 9 be indicated allowable.

2) AAPA, Uri and Shuler do not teach "decreasing usage of said computing resource by said workload...said decreasing usage does not halt operation of said workload" as recited in independent claim 9

The Final Office Action cited the combination of AAPA, Uri and Shuler as teaching the step of "decreasing usage of said computing resource by said workload...said decreasing usage does not halt operation of said workload." Appellant respectfully disagrees with the Examiner's assertion. First, AAPA and Uri do not disclose "said decreasing usage does not halt operation of

said workload” as recited in independent claim 9 and the Examiner admitted that AAPA and Uri do not teach this feature. See top of page 4 of Final Office Action.

Second, Appellant asserts that Shuler does not overcome this deficiency. The Examiner asserted that Shuler teaches an improvement in garbage collecting where halting of processing would typically occur, the improvement including garbage collecting that remedies delays and does not halt operation of the processor (e.g., a “list processor” that manipulates linked list structures). See Final Office Action at middle of page 4 and bottom of page 14. Further, the Examiner asserted that it would have been obvious to modify AAPA in view of Uri with the teachings of Shuler such that the decreasing does not halt operation in order to increase the responsiveness of AAPA by continually executing the workload while performing memory paging. See Final Office Action at middle of page 4. Appellant respectfully disagrees that Shuler discloses “decreasing usage of said computing resource by said workload...said decreasing usage does not halt operation of said workload” as recited in independent claim 9 and, as a result, the combined teaching of these three references fails to show or suggest this limitation of independent claim 9.

The above discussed step of “decreasing usage of said computing resource by said workload...said decreasing usage does not halt operation of said workload” essentially calls for each workload whose resource usage exceeds its respective allotment to remains *partially operable* (e.g., between full operation and halted operation) during the paging out of its respective range of computer resources. Here, the Examiner admitted that the combination of AAPA and Uri fails to disclose not halting or in other words fails to disclose continuing operation of the workload, i.e., the combination of AAPA and Uri discloses halting operation or teaches directly away from the claimed invention. See top of page 4 of Final Office Action.

However, Shuler merely discloses either halting list processor operation (“Processing must be halted,” see column 1, line 56) or full operation of the list processor (“...without halting of the list processor...,” “...list processing need never halt...,” see column 2, lines 1-11). In this regard, Shuler never discloses or suggests decreasing usage of the workload without halting usage of the workload (e.g., partial operation of the workload) as recited in independent claim 9.

This feature advantageously allows a workload's resource consumption to be reduced while maintaining partial operation of the workload which beneficially continues operation and responds to actions of the computer system. See Appellant's specification at page 25, lines 1-3. For these additional reasons, Appellant respectfully submits that AAPA, Uri and Shuler do not disclose or suggest "decreasing usage of said computing resource by said workload...said decreasing usage does not halt operation of said workload" as recited in independent claim 9 and requests that this rejection be overturned by the Board.

3) Shuler teaches away from the method of independent claim 9 because Shuler's garbage collection mechanism runs continuously as opposed "responsive to exceeding a limit on utilization of said computing resource" as recited in independent claim 9

On the top of page 15 of the Final Office Action (the "Response to Arguments" section), the Examiner contended that as "Shuler teaches that such techniques (e.g., the mark and sweep techniques) require up to three times as much processing power (column 2, lines 7-11)", it necessarily follows that "garbage collection is run when called and not continuously or requiring fixed overhead of continual paging as argued by Applicant." Appellant disagrees with the Examiner's conclusion. The portion cited by the Examiner only discloses that three times as much processing power may be needed to *guarantee that the list processing need never halts*. Stated otherwise, the portions cited by the Examiner provide no indication that garbage collection occurs only when called or needed. Rather, one could just as easily conclude that the garbage collection occurs *continuously* from the use of three times the processing power as the Examiner's speculation. As the disclosure of Shuler therefore teaches away from the method of independent claim 9, Appellant respectfully requests that the Board overturn this rejection for this additional reason.

4) The background section of Shuler cannot be combined with the primary invention of Shuler to collectively define the "teaching" of Shuler because doing so would destroy the primary invention of Shuler

With reference to the "Response to Arguments" section of the Final Office Action (e.g., paragraphs 52 and 53), the Examiner contended that even though the garbage collection process of Shuler is performed on a continual basis (and is thus *distinct* from the methods of independent claims 9, 16, 36 and 38 which decrease computing resource usage and/or initiate a paging out

process *in response to* a computer resource usage limit being exceeded), the background of Shuler discusses “various other prior art techniques” and that Shuler’s invention along with the discussed “prior art techniques” collectively define the teaching of Shuler. Appellant respectively disagrees with the Examiner’s assertion.

Even assuming that Shuler did disclose or suggest decreasing usage or initiating a paging our procedure *in response to a computing resource limit being exceeded* (e.g., not on a continual basis), modifying the primary invention of Shuler (a reference counter garbage collection mechanism for list processing whereby there is never an unanticipated delay when needing a free cell, see column 3, lines 30-35 and column 4, lines 11-13) so that it only operates in response to, for instance, a computing resource limit being exceeded, would destroy a primary objective and intended purpose of the invention of Shuler (e.g., to operate such that there is never an unanticipated delay, see “Summary of the Invention” at column 4, lines 11-13). It is only with hindsight reasoning that the Examiner has alleged that the method of independent claim 9 is somehow obvious in view of the cited references.

5) Combining AAPA, Uri and Shuler would result in an invention distinct from the method of independent claim 9

Combining AAPA, Uri and Shuler would result in a method of implementing a process of paging out pages of a workload from memory (from AAPA) that is implemented as an “in-kernel” process with memory management subsystems for various groups (from Uri) whereby the process is either fully operational (e.g., does not halt) or else halts completely (from Shuler). Conversely, the method of independent claim 9, inter alia, monitors workload computing resource usage via a process within a user space and decreases workload computing resource usage in response to exceeding a utilization limit in a manner that does not halt operation of the workload. As AAPA, Uri and Shuler do not either alone or in any combination disclose all the elements of independent claim 9, Appellant respectfully requests that independent claim 9 be indicated allowable.

Claims 12, 14 and 15 depend from independent claim 9 and therefore are allowable for those reasons supporting the allowability of independent claim 9. Furthermore, claims 12, 14 and 15 recite additional features not disclosed by AAPA, Uri and Shuler. For instance, claim 14

recites “wherein said decreasing usage is initiated by a process of said workload,” and the Examiner asserted that AAPA discloses this feature at page 5, lines 18-20 of Appellant’s specification. However, this portion merely discloses that “processes may be combined for the purposes of monitoring and controlling memory usage, in order to limit the number of entities monitored.” Furthermore, six lines prior in the same paragraph, it is disclosed that “this method is generally implemented as an operating system kernel process.” Thus, the “processes” that are combined are implemented *as an operating system kernel process* which is distinct from being “initiated by a process of said workload” as recited in claim 14. For this additional reason, Appellant respectfully requests that claim 14 be indicated allowable.

Independent Claim 16:

Independent claim 16 is directed to a computer implemented method for memory management of a workload from within the workload. See representative computer system in Figure 1 and flow diagram of Figure 4. The method includes accessing a list of memory pages assigned to said workload in a user space, the workload includes a plurality of running processes, the plurality of running processes are a subset of all processes that are running in the user space. See box 410 in Figure 4. Thereafter, “responsive to a request from a first process of said workload for memory which exceeds a predetermined memory limit for said workload, selecting a plurality of memory pages from said list of memory pages, wherein the plurality of memory pages includes least recently used memory pages assigned to the workload.” See box 420 in Figure 4 and page 23, lines 8-12. The method also includes “initiating a second process within the user space to page out said plurality of memory pages wherein at least a portion of said workload continues to operate subsequent to said initiating.” See box 430 in Figure 4. AAPA, Uri and Shuler do not alone or in any combination disclose or suggest the method of independent claim 16.

1) AAPA, Uri and Shuler do not teach “initiating a second process within the user space to page out said plurality of memory pages” as recited in independent claim 16

As admitted by the Examiner, AAPA does not at least disclose or suggest “initiating a second process within the user space to page out said plurality of memory pages” as recited in independent claim 16. See bottom page 5 of the Final Office Action. To remedy this deficiency of AAPA, the Examiner cited to page 3, paragraph 2 of Uri which discloses the aforementioned

“Memory Resource Groups” wherein a block of memory is allocated to each PRM group or WLM workload and a separate memory management subsystem will be created for each group, and asserted that it would have been obvious that the “accessing” is performed within the user space to allow for more fine grained control and for resources to be allocated based on business priorities. See top half of page 6 of the Final Office Action. For purposes of this Appeal Brief, Appellant will assume that the “accessing” referred to by the Examiner is the same as the “initiating a second process” as recited in independent claim 16. Appellant nevertheless respectfully disagrees with the Examiner’s assertion.

Actually, the Memory Resource Groups and memory management subsystems are implemented as *in-kernel memory controls*. See paragraph 2, lines 10-11 of Uri. In contrast, independent claim 16 recites that the initiating a second process to page out the memory pages *is performed by a process within a user space*. Thus, the system of Uri provides an opposite technique from the method of independent claim 16 at least in this regard. The combination of AAPA and Uri fail to suggest or show the “initiating” step and Shuler is not cited for any teaching relevant to the initiating step. For at least this reason, Appellant respectfully requests that independent claim 16 be indicated allowable.

2) AAPA, Uri and Shuler do not teach “wherein at least a portion of said workload continues to operate subsequent to said initiating” (e.g., remains partially operable) as recited in independent claim 16

As admitted by the Examiner, AAPA and Uri do not disclose or suggest this limitation. See middle of page 6 of the Final Office Action. To remedy this failure of AAPA and Uri, the Examiner contended that Shuler teaches an improvement in garbage collecting where halting of processing would typically occur, the improvement including garbage collecting that remedies delays and does not halt operation of the processor, and that it would have been obvious to modify the teaching of AAPA and Uri such that decreasing usage does not halt operation to increase responsiveness of AAPA. See bottom of page 6 of the Final Office Action. Appellant respectfully disagrees with the Examiner’s proposed combination.

As discussed above with respect to independent claim 9 (see previous discussion), Shuler never discloses or suggests allowing a workload to “continue to operate subsequent to said

initiating” (e.g., the workload remains partially operable), and therefore combining Shuler with AAPA and Uri would not disclose or suggest this limitation. Moreover, Shuler therefore also cannot disclose or suggest doing so “responsive to a request from a first process of said workload for memory which exceeds a predetermined memory limit for said workload” as recited in independent claim 16. Also see previous discussion in relation to independent claim 9. For all of the above reasons, Appellant respectfully requests that the Board overturns these rejections and that independent claim 16 be indicated allowable.

Further, Claims 17-23 and 25 are allowable for the same reasons put forth in support of the allowability of independent claim 16 and recite additional features not disclosed by AAPA, Uri and Shuler. For instance, claim 18 recites “wherein said second process is not an operating system kernel process,” and the Examiner admitted that AAPA, Uri and Shuler were all deficient in this regard. See top half of page 7 of Final Office Action. However, the Examiner asserted that as AAPA discloses that “operating system kernel processes can be very expensive in terms of computing load on a system as they execute very frequently” on lines 13-15 of page 5 of Appellant’s specification, “it would have been obvious...to perform the monitoring by some other process.” Appellant respectfully disagrees with the Examiner’s conclusion.

With continued reference to page 5 of Appellant’s specification, lines 16-17 actually disclose that as a consequence of the computing load expense of operating system kernel processes, “it is generally necessary to limit such methods to monitoring a very few processes.” Thus, contrary to the Examiner’s conjecture, the typical response to the high computing load expense of operating system kernel processes is to limit such methods to monitoring a very few processes as opposed to initiating a second process *that is not an* operating system kernel process as in claim 18. For this additional reason, Appellant respectfully requests that claim 18 be indicated allowable.

Independent Claim 38:

Independent claim 38 is directed to a computer implemented method of managing computer resources over a plurality of workloads that includes, for each workload of said plurality of workloads, monitoring respective workload resource usage against a respective

allotment of each workload. A representative computer system 100 usable in conjunction with the method of independent claim 38 is shown in Figure 1 that illustrates a user space 110 with one or more workloads 111-115, each of which may include one or more processes 121-123 and may be allocated computing resources (e.g., physical memory). See page 13, line 17 through page 14, line 16 of Appellant's specification. The method also includes determining a range of computer resources to page out for each workload whose resource usage exceeds its respective allotment. For instance, a plurality of "least recently-used pages" may be paged out. See page 15, line 22 through page 16, line 1. Moreover, the method includes initiating a paging out operation of said range of computer resources and wherein said monitoring, said determining and said initiating all occur within a process of user space. See box 430 of Figure 4, page 13, lines 2-4, page 15, lines 9-21 and page 16, lines 1-3.

Further, each of the plurality of workloads exists within a user space and includes a plurality of running processes, the plurality of running processes are a subset of all processes that are running in the user space, and paging out said range of computer resources. As seen in Figure 1, the workload 112 includes a plurality of running processes 121-123 which are a subset of all processes running (e.g., processes 121, 122 running in other workloads 111, 113-115) in the user space 110. Moreover, each workload whose resource usage exceeds its respective allotment remains partially operable during said paging out of its respective range of computer resources. AAPA, Uri and Shuler do not alone or in any combination disclose or suggest the method of independent claim 38.

AAPA, Uri and Shuler do not teach "wherein said monitoring, said determining and said initiating all occur within a process of user space" and "wherein each workload whose resource usage exceeds its respective allotment remains partially operable during said paging out of its respective range of computer resources" as recited in independent claim 38

As these features are similar to those discussed previously with respect to independent claims 9 and/or 16, Appellant respectfully asserts that independent claim 38 and its respective dependent claims (e.g., claims 39 and 40) be indicated allowable for those reasons as previously discussed.

Rejection of Claims 36 and 37 under 35 U.S.C. §103(a) Based on Tanenbaum, AAPA and Shuler is Improper

Independent Claim 36:

Independent claim 36 is directed to a computer implemented method including accessing memory usage for a workload and examining page usage for each process of the workload, the workload existing within a user space and including a plurality of running processes that are a subset of all processes that are running in the user space. The method also includes aggregating usage of the each process to determine an aggregate usage for the workload. See box 610 in Figure 6. Thereafter, if the aggregate usage does not exceed a memory utilization limit for the workload (see diamond 620), the method includes repeating the accessing and aggregating for a next workload. See box 625 in Figure 6. However, if the aggregate usage exceeds the memory utilization limit for the workload, the method includes determining least recently used pages based on accessed bits associated with the workload. See box 630 in Figure 6. Additionally, if the aggregate usage exceeds the memory utilization limit for the workload, the method includes supplying a range of least recently used pages in a system call to an operating system kernel for evicting the range of least recently used pages to reduce resource usage by the workload. See box 640 in Figure 6 and page 27, lines 1-3. The method also includes retaining at least partial operation of said workload during said page evicting such that the at least partial operation does not stop said workload. See box 650 in Figure 6 and page 27, lines 5-6. Tanenbaum, AAPA, and Shuler do not alone or in any combination disclose or suggest the method of independent claim 36.

1) Tanenbaum, AAPA and Shuler do not teach “retaining at least partial operation of said workload during said page eviction such that the at least partial operation does not stop said workload” as recited in independent claim 36

The portion of Tanenbaum cited by the Examiner at the bottom half of page 11 of the Final Office Action (e.g., page 234, section 4.6.1, “Local versus Global Allocation Policies”) is directed to how memory should be allocated among competing runnable processes (e.g., processes A, B and C, each including a number of pages in memory) when a page fault occurs. More specifically and in the scenario where process A gets a page fault, “local” page

replacement considers only least recently used pages in process A whereas “global” page replacement considers the least recently used pages of processes A, B and C as a whole.

As admitted by the Examiner, Tanenbaum and AAPA do not disclose “retaining at least partial operation of said workload during said page eviction such that the at least partial operation does not stop said workload” as recited in independent claim 36. See bottom of page 12 of the Final Office Action. The Examiner turned to Shuler supply a teaching for this feature. However and as discussed above (e.g., in relation to independent claim 16), Shuler also does not disclose this feature and as a result, the combination of Tanenbaum, AAPA and Shuler cannot disclose this feature. Therefore, independent claim 36 is allowable for at least this reason and such disposition is respectfully requested.

2) Tanenbaum, AAPA and Shuler do not teach the determination of least recently used pages “if said aggregate usage exceeds a memory utilization limit for a workload” as recited in independent claim 36.

As admitted by the Examiner, Tanenbaum does not disclose this feature. See top of page 12 of Final Office Action. Furthermore, Appellant notes that Shuler was not discussed by the Examiner in relation to this feature and that Shuler does not otherwise disclose this feature. However, the Examiner asserted that “AAPA teaches paging out memory in response to workloads exceeding their respective allotments,” and that “it would have been obvious...to perform the determination in response to an aggregate usage exceeding the memory utilization limit...to ensure that overall memory usage does not exceed the amount that is physically available.” Appellant respectfully disagrees with the Examiner’s proposed combination.

As discussed above, the portion of Tanenbaum cited by the Examiner discusses how memory should be allocated *when a page fault occurs*, namely, when a program accesses a page that is mapped in address space, but not loaded in physical memory. However, modifying the memory allocation process of Tanenbaum to occur in response to an aggregate usage exceeding a memory utilization limit as proposed by the Examiner would render the memory allocation process inoperable for its intended purpose and thus render’s the Examiner’s combination improper. More specifically, the memory allocation process may result in deadlock, for instance, when process A has experienced a page fault and process B has exceeded a memory utilization


limit because it does not appear that the memory allocation process is designed to decide whether, for instance, to consider only the least recently used pages of process A or those of process B. For all of the above reasons, Appellant respectfully requests that independent claim 36 (and dependent claim 37) be indicated allowable.

Conclusion

In view of the above remarks, the pending claims are believed allowable and the case in condition for allowance. Appellant respectfully requests that the rejections of all pending claims be reversed. Appellant has met or exceeded the burden of overcoming the prima facie case of unpatentability made out by the Examiner by providing rebuttal evidence of adequate weight and explaining how and why the record indicates the pending claims are patentable.

Respectfully submitted,

Date: JANUARY 5, 2009


Jonathon A. Szumny, Reg. No. 57,695
Marsh Fischmann & Breyfogle LLP
8055 E. Tufts Ave., Suite 450
Denver, CO 80237
Phone: (303) 770-0051
Fax: (303) 770-0152

VIII. CLAIMS APPENDIX

9. A computer implemented method comprising:

monitoring usage of a computing resource utilized by a workload wherein said monitoring is performed by a process within a user space and the process monitors the user space only, the workload includes a plurality of running processes, the plurality of running processes are a subset of all processes that are running in the user space;

responsive to exceeding a limit on utilization of said computing resource, decreasing usage of said computing resource by said workload, said computing resource comprises physical memory and said decreasing usage of said computing resource comprises paging a portion of said physical memory assigned to said workload out of said physical memory and said decreasing usage does not halt operation of said workload.

12. The method of claim 9 wherein said portion of said physical memory comprises a least recently used portion of said physical memory assigned to said workload.

14. The method of claim 9 wherein said decreasing usage is initiated by a process of said workload.

15. The method of claim 9 wherein said process that performs said monitoring is not an operating system kernel process.

16. A computer implemented method for memory management of a workload from within the workload comprising:

accessing a list of memory pages assigned to said workload in a user space, the workload includes a plurality of running processes, the plurality of running processes are a subset of all processes that are running in the user space;

responsive to a request from a first process of said workload for memory which exceeds a predetermined memory limit for said workload, selecting a plurality of memory pages from said list of memory pages, wherein the plurality of memory pages includes least recently used memory pages assigned to the workload; and

initiating a second process within the user space to page out said plurality of memory pages wherein at least a portion of said workload continues to operate subsequent to said initiating.

17. The method of claim 16 wherein said accessing, selecting and initiating are performed by said second process within said workload.

18. The method of claim 16 wherein said second process is not an operating system kernel process.

19. The method of claim 16 wherein said second process is loaded into a user space.

20. The method of claim 16 wherein said plurality of memory pages comprises memory pages that are least recently used.

21. The method of claim 20 wherein said plurality of memory pages comprises memory pages that are least recently used by said workload.

22. The method of claim 20 wherein said page out of said plurality of least recently used memory pages reduces a number of memory pages assigned to said workload to below said memory limit.

23. The method of claim 20 wherein said plurality of least recently used memory pages comprises the minimum number of memory pages to reduce said number of memory pages assigned to said workload below said memory limit.

25. The method of claim 16 wherein said initiating is not performed by an operating system kernel process.

36. A computer implemented method comprising:

accessing memory usage for a workload and examining page usage for each process of said workload, the workload exists within a user space and includes a plurality of running processes, the plurality of running processes are a subset of all processes that are running in the user space;

aggregating usage of said each process to determine an aggregate usage for said workload;

if said aggregate usage does not exceed a memory utilization limit for said workload, repeating said accessing and aggregating for a next workload;

if said aggregate usage exceeds said memory utilization limit for said workload, determining least recently used pages based on accessed bits associated with said workload;

if said aggregate usage exceeds said memory utilization limit for said workload, supplying a range of least recently used pages in a system call to an operating system kernel for evicting said range of least recently used pages to reduce resource usage by said workload; and

retaining at least partial operation of said workload during said page evicting such that the at least partial operation does not stop said workload.

37. The method of claim 36 wherein said determining and said supplying occur in a plurality of user space processes.

38. A computer implemented method of managing computer resources over a plurality of workloads, said method comprising:

for each workload of said plurality of workloads, monitoring respective workload resource usage against a respective allotment of each workload;

determining a range of computer resources to page out for each workload whose resource usage exceeds its respective allotment; and

initiating a paging out operation of said range of computer resources and wherein said monitoring, said determining and said initiating all occur within a process of user space,

wherein each of the plurality of workloads exists within a user space and includes a plurality of running processes, the plurality of running processes are a subset of all processes that are running in the user space, and paging out said range of computer resources and wherein each workload whose resource usage exceeds its respective allotment remains partially operable during said paging out of its respective range of computer resources.

39. The method as described in claim 38 wherein said determining comprises determining least recently used pages for each workload whose resource usage exceeds its respective allotment.

40. The method as described in claim 38 wherein said process is situated within a workload of said plurality of workloads.

IX. EVIDENCE APPENDIX

No copies of evidence are required with this Appeal Brief. Appellant has not relied upon any evidence submitted under 37 C.F.R. §§ 1.130, 1.131, or 1.132.

X. RELATED PROCEEDINGS APPENDIX

There are no copies of decisions rendered by a court or the Board to provide with this Appeal as there are no related proceedings.